OIPE
JUL 0 8 2004
PATENT & TRADEMARK OFFICE

Serial No.: 10/764,137
Inventor(s): Constantin Chassapis, Sven K. Esche, Jan W. Nazalewicz, Dennis J. Hromin
Title: Multi-User, Multi-Device Remote Access System

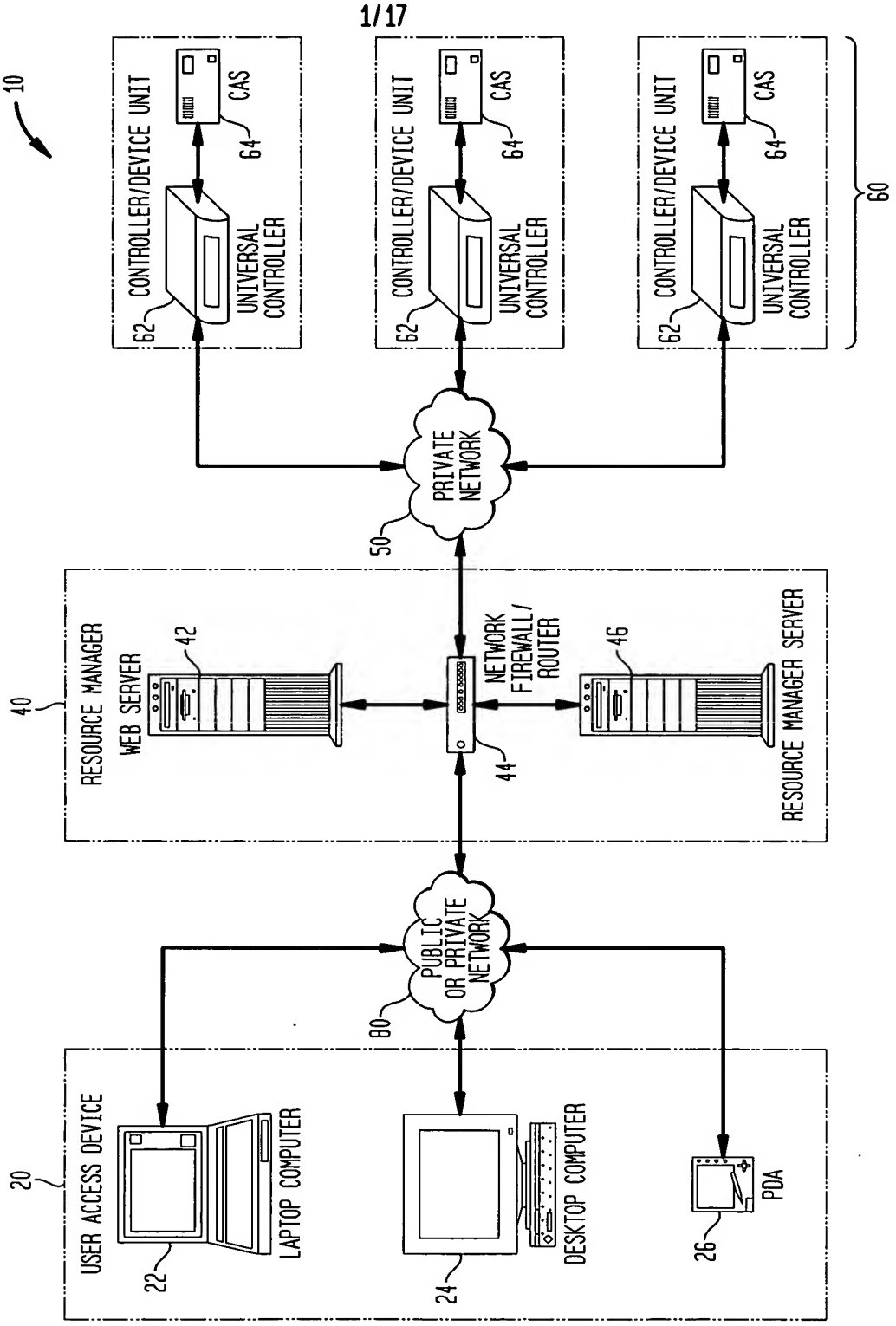


FIG. 1

FIG. 2

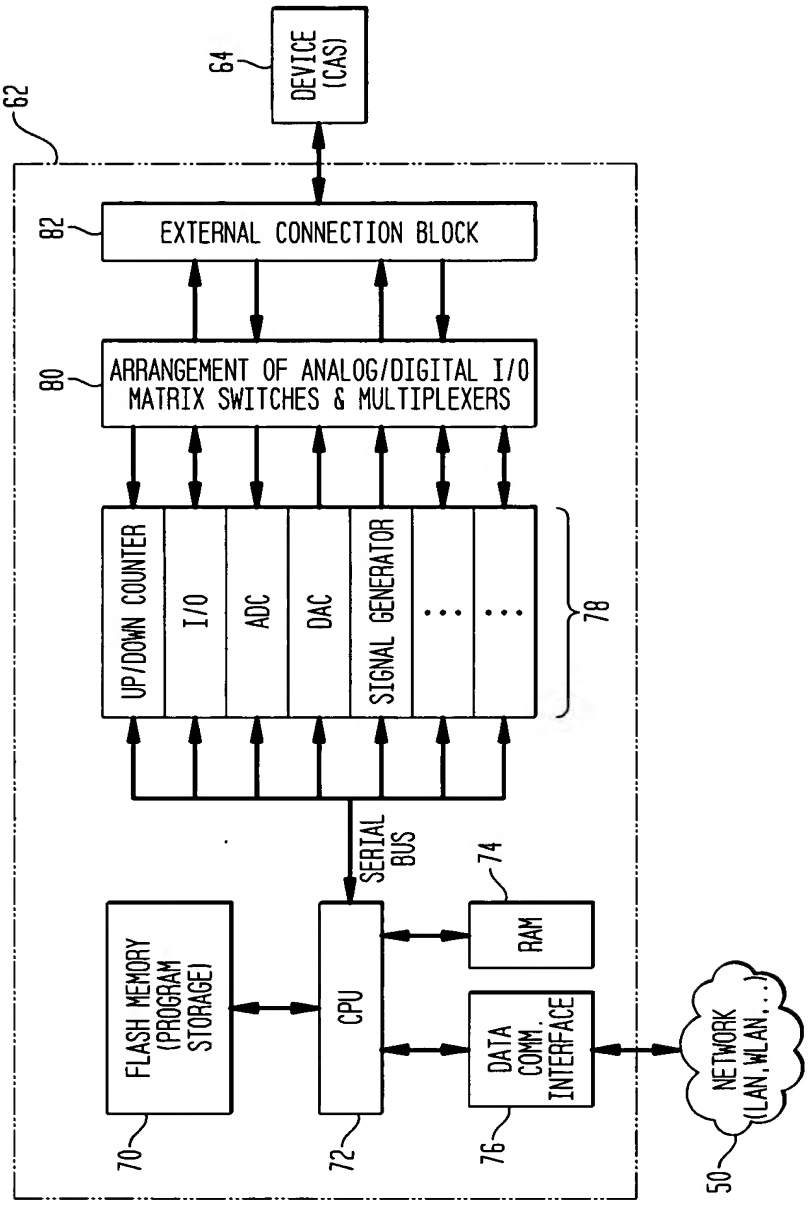

64
DEVICE (CAS)
62
82
EXTERNAL CONNECTION BLOCK
80
ARRANGEMENT OF ANALOG/DIGITAL I/O MATRIX SWITCHES & MULTIPLEXERS
UP/DOWN COUNTER
I/O
ADC
DAC
SIGNAL GENERATOR
78
SERIAL BUS
74
RAM
FLASH MEMORY (PROGRAM STORAGE)
CPU
DATA COMM. INTERFACE
NETWORK (LAN, WLAN, ...)
70
72
76
50

FIG. 3

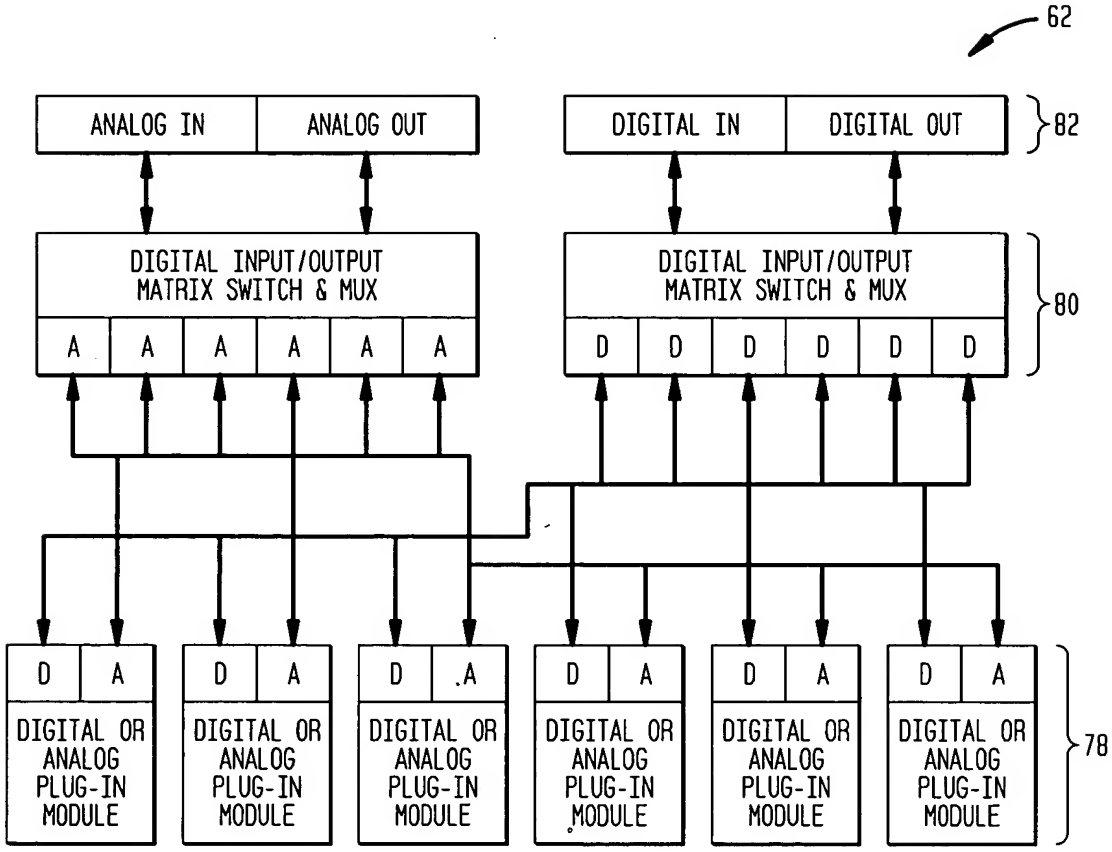

62
ANALOG IN
ANALOG OUT
DIGITAL IN
DIGITAL OUT
82
DIGITAL INPUT/OUTPUT MATRIX SWITCH & MUX
A A A A A A
DIGITAL INPUT/OUTPUT MATRIX SWITCH & MUX
D D D D D D
80
D A
DIGITAL OR ANALOG PLUG-IN MODULE
78

FIG. 4

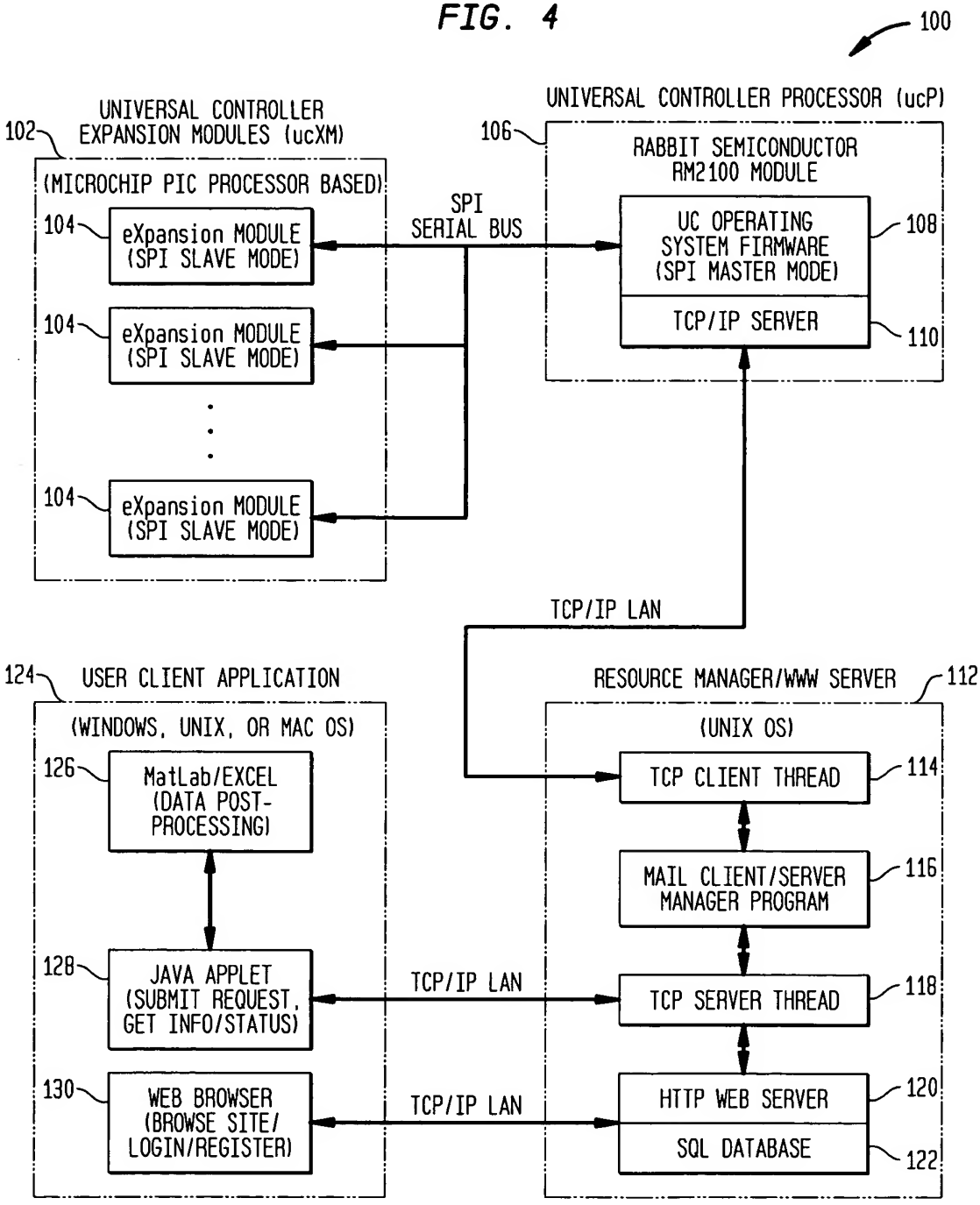

100
UNIVERSAL CONTROLLER EXPANSION MODULES (ucXM)
102
(MICROCHIP PIC PROCESSOR BASED)
104
eXpansion MODULE (SPI SLAVE MODE)
104
eXpansion MODULE (SPI SLAVE MODE)
104
eXpansion MODULE (SPI SLAVE MODE)
SPI SERIAL BUS
UNIVERSAL CONTROLLER PROCESSOR (ucP)
106
RABBIT SEMICONDUCTOR RM2100 MODULE
UC OPERATING SYSTEM FIRMWARE (SPI MASTER MODE)
108
TCP/IP SERVER
110
TCP/IP LAN
124
USER CLIENT APPLICATION
(WINDOWS, UNIX, OR MAC OS)
126
MatLab/EXCEL (DATA POST-PROCESSING)
128
JAVA APPLET (SUBMIT REQUEST, GET INFO/STATUS)
130
WEB BROWSER (BROWSE SITE/ LOGIN/REGISTER)
TCP/IP LAN
TCP/IP LAN
RESOURCE MANAGER/WWW SERVER
112
(UNIX OS)
TCP CLIENT THREAD
114
MAIL CLIENT/SERVER MANAGER PROGRAM
116
TCP SERVER THREAD
118
HTTP WEB SERVER
120
SQL DATABASE
122

FIG. 5

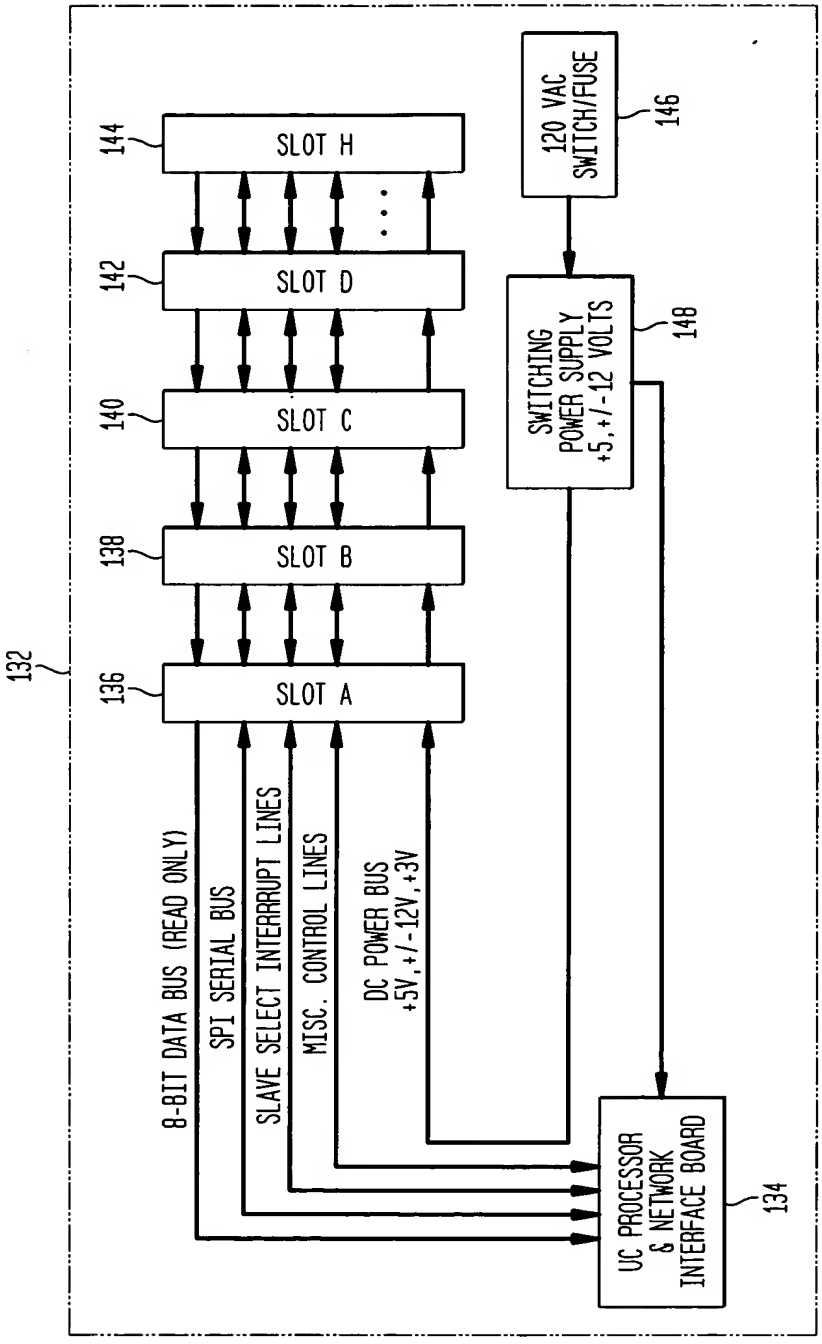

132
144
SLOT H
142
SLOT D
140
SLOT C
138
SLOT B
136
SLOT A
120 VAC SWITCH/FUSE
146
SWITCHING POWER SUPPLY +5,+/-12 VOLTS
148
8-BIT DATA BUS (READ ONLY)
SPI SERIAL BUS
SLAVE SELECT INTERRRUPT LINES
MISC. CONTROL LINES
DC POWER BUS +5V,+/-12V,+3V
UC PROCESSOR & NETWORK INTERFACE BOARD
134

FIG. 6

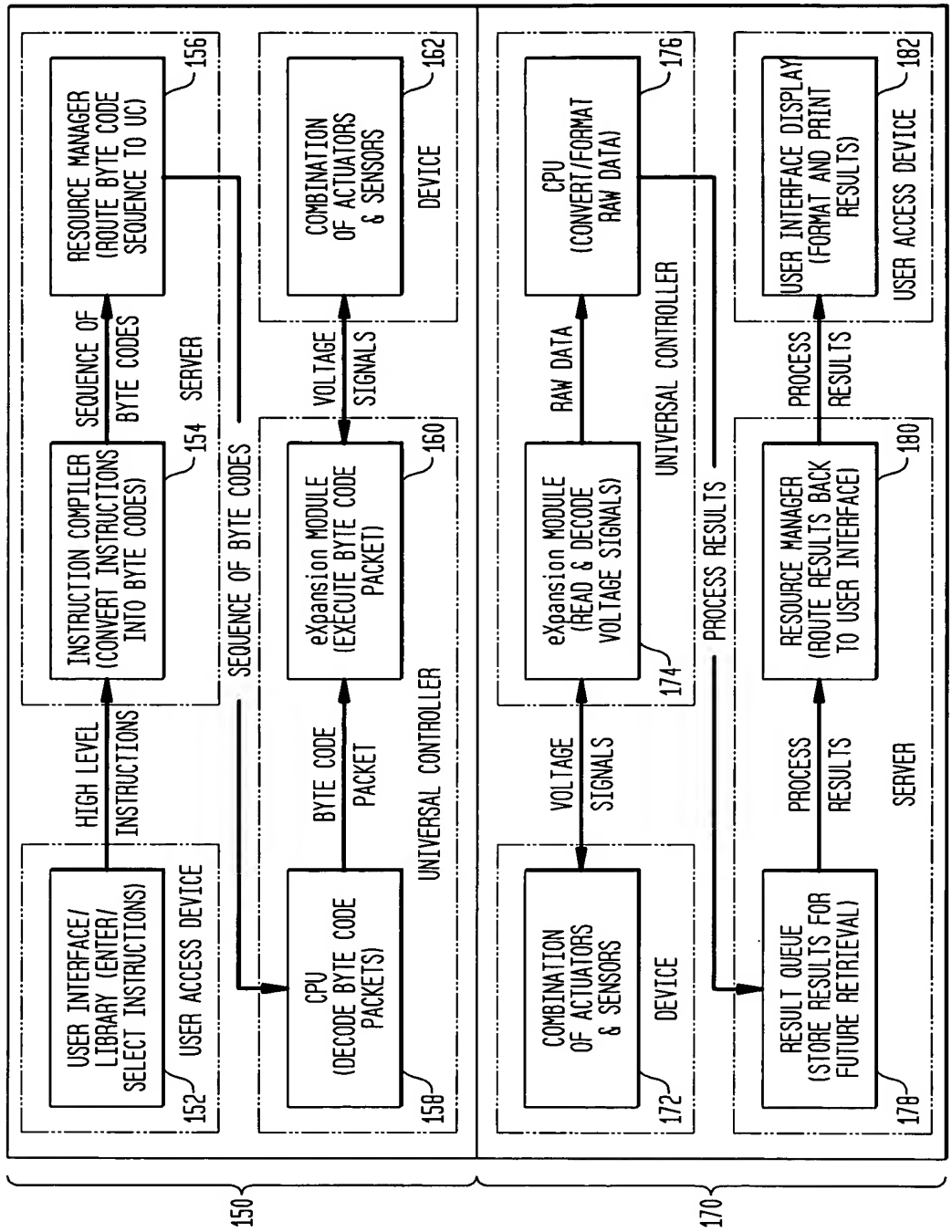

USER INTERFACE/ LIBRARY (ENTER/ SELECT INSTRUCTIONS)
152
USER ACCESS DEVICE
HIGH LEVEL INSTRUCTIONS
INSTRUCTION COMPILER (CONVERT INSTRUCTIONS INTO BYTE CODES)
154
SERVER
SEQUENCE OF BYTE CODES
RESOURCE MANAGER (ROUTE BYTE CODE SEQUENCE TO UC)
156
SEQUENCE OF BYTE CODES
CPU (DECODE BYTE CODE PACKETS)
158
BYTE CODE PACKET
eXpansion MODULE (EXECUTE BYTE CODE PACKET)
160
UNIVERSAL CONTROLLER
VOLTAGE SIGNALS
COMBINATION OF ACTUATORS & SENSORS
162
DEVICE
150
COMBINATION OF ACTUATORS & SENSORS
172
DEVICE
VOLTAGE SIGNALS
eXpansion MODULE (READ & DECODE VOLTAGE SIGNALS)
174
RAW DATA
CPU (CONVERT/FORMAT RAW DATA)
176
UNIVERSAL CONTROLLER
PROCESS RESULTS
RESULT QUEUE (STORE RESULTS FOR FUTURE RETRIEVAL)
178
SERVER
PROCESS RESULTS
RESOURCE MANAGER (ROUTE RESULTS BACK TO USER INTERFACE)
180
PROCESS RESULTS
USER INTERFACE DISPLAY (FORMAT AND PRINT RESULTS)
182
USER ACCESS DEVICE
170

FIG. 7A

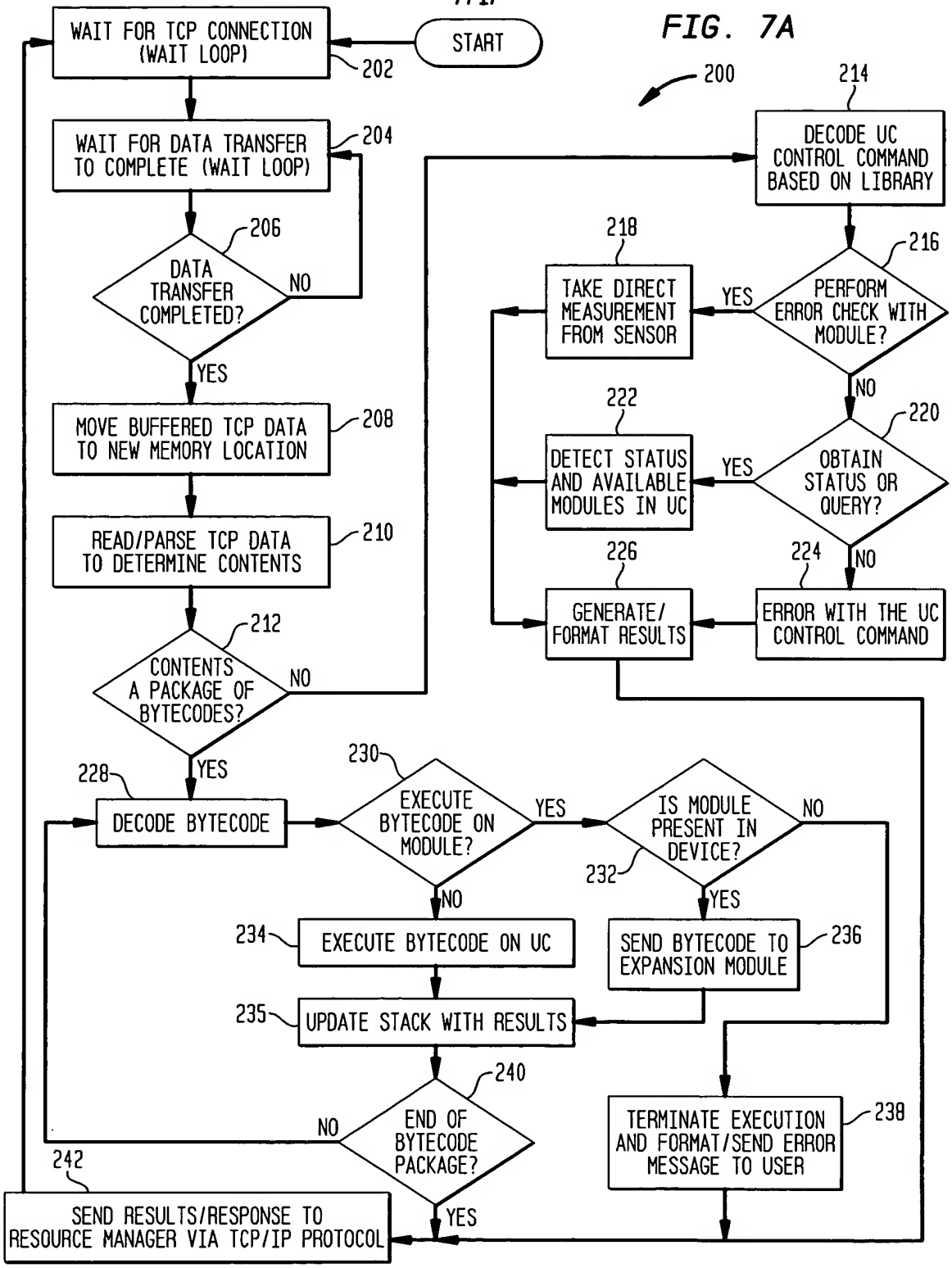

START
WAIT FOR TCP CONNECTION (WAIT LOOP)
202
WAIT FOR DATA TRANSFER TO COMPLETE (WAIT LOOP)
204
DATA TRANSFER COMPLETED?
206
NO
YES
MOVE BUFFERED TCP DATA TO NEW MEMORY LOCATION
208
READ/PARSE TCP DATA TO DETERMINE CONTENTS
210
CONTENTS A PACKAGE OF BYTECODES?
212
NO
YES
DECODE BYTECODE
228
EXECUTE BYTECODE ON MODULE?
230
YES
NO
IS MODULE PRESENT IN DEVICE?
232
NO
YES
EXECUTE BYTECODE ON UC
234
SEND BYTECODE TO EXPANSION MODULE
236
UPDATE STACK WITH RESULTS
235
END OF BYTECODE PACKAGE?
240
NO
YES
TERMINATE EXECUTION AND FORMAT/SEND ERROR MESSAGE TO USER
238
SEND RESULTS/RESPONSE TO RESOURCE MANAGER VIA TCP/IP PROTOCOL
242
200
DECODE UC CONTROL COMMAND BASED ON LIBRARY
214
PERFORM ERROR CHECK WITH MODULE?
216
YES
TAKE DIRECT MEASUREMENT FROM SENSOR
218
NO
OBTAIN STATUS OR QUERY?
220
YES
DETECT STATUS AND AVAILABLE MODULES IN UC
222
NO
ERROR WITH THE UC CONTROL COMMAND
224
GENERATE/ FORMAT RESULTS
226

FIG. 7B

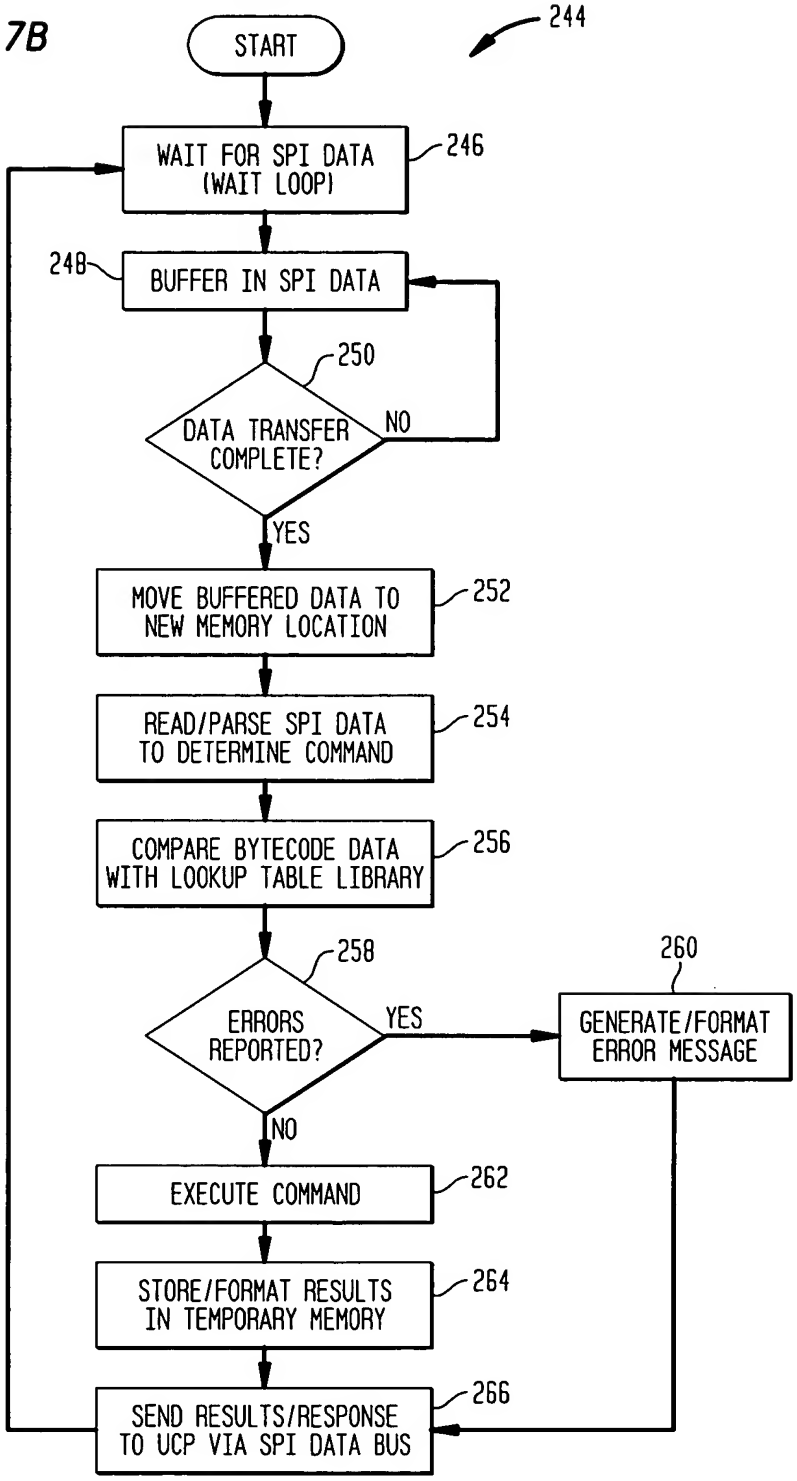

244
START
WAIT FOR SPI DATA (WAIT LOOP)
246
248
BUFFER IN SPI DATA
250
DATA TRANSFER COMPLETE?
NO
YES
MOVE BUFFERED DATA TO NEW MEMORY LOCATION
252
READ/PARSE SPI DATA TO DETERMINE COMMAND
254
COMPARE BYTECODE DATA WITH LOOKUP TABLE LIBRARY
256
258
ERRORS REPORTED?
YES
260
GENERATE/FORMAT ERROR MESSAGE
NO
EXECUTE COMMAND
262
STORE/FORMAT RESULTS IN TEMPORARY MEMORY
264
SEND RESULTS/RESPONSE TO UCP VIA SPI DATA BUS
266

FIG. 8

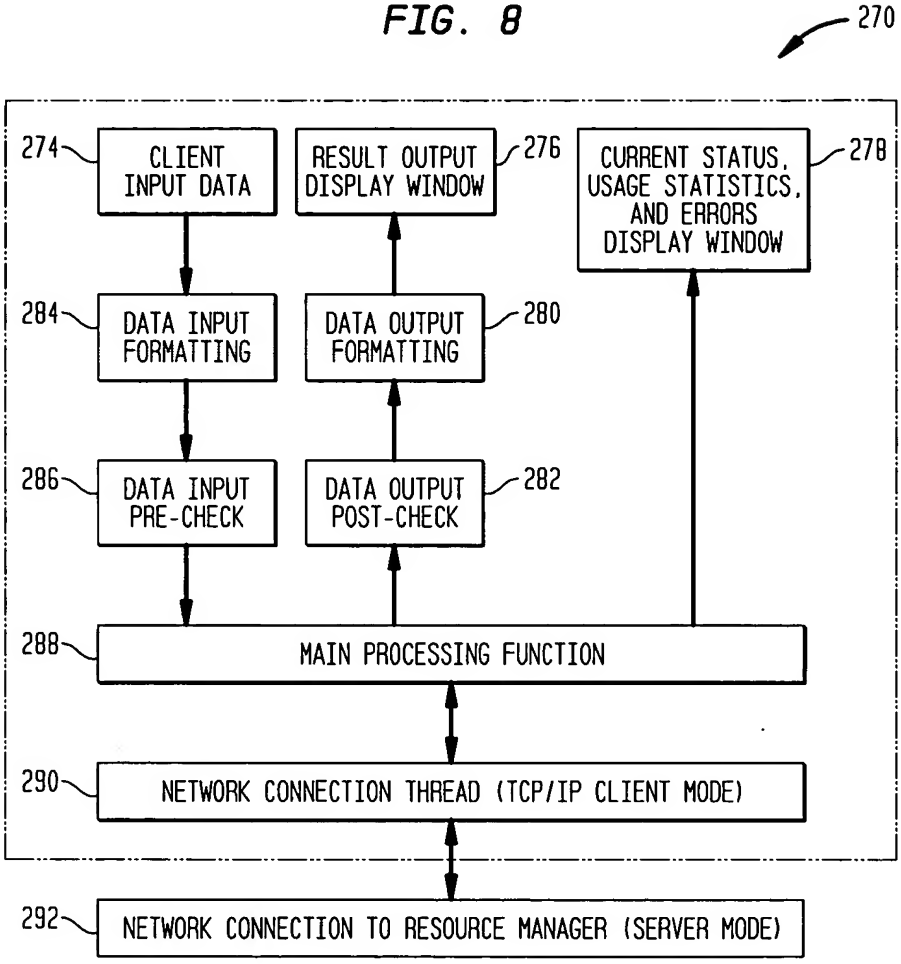

270
274
CLIENT INPUT DATA
276
RESULT OUTPUT DISPLAY WINDOW
278
CURRENT STATUS, USAGE STATISTICS, AND ERRORS DISPLAY WINDOW
284
DATA INPUT FORMATTING
280
DATA OUTPUT FORMATTING
286
DATA INPUT PRE-CHECK
282
DATA OUTPUT POST-CHECK
288
MAIN PROCESSING FUNCTION
290
NETWORK CONNECTION THREAD (TCP/IP CLIENT MODE)
292
NETWORK CONNECTION TO RESOURCE MANAGER (SERVER MODE)

FIG. 9

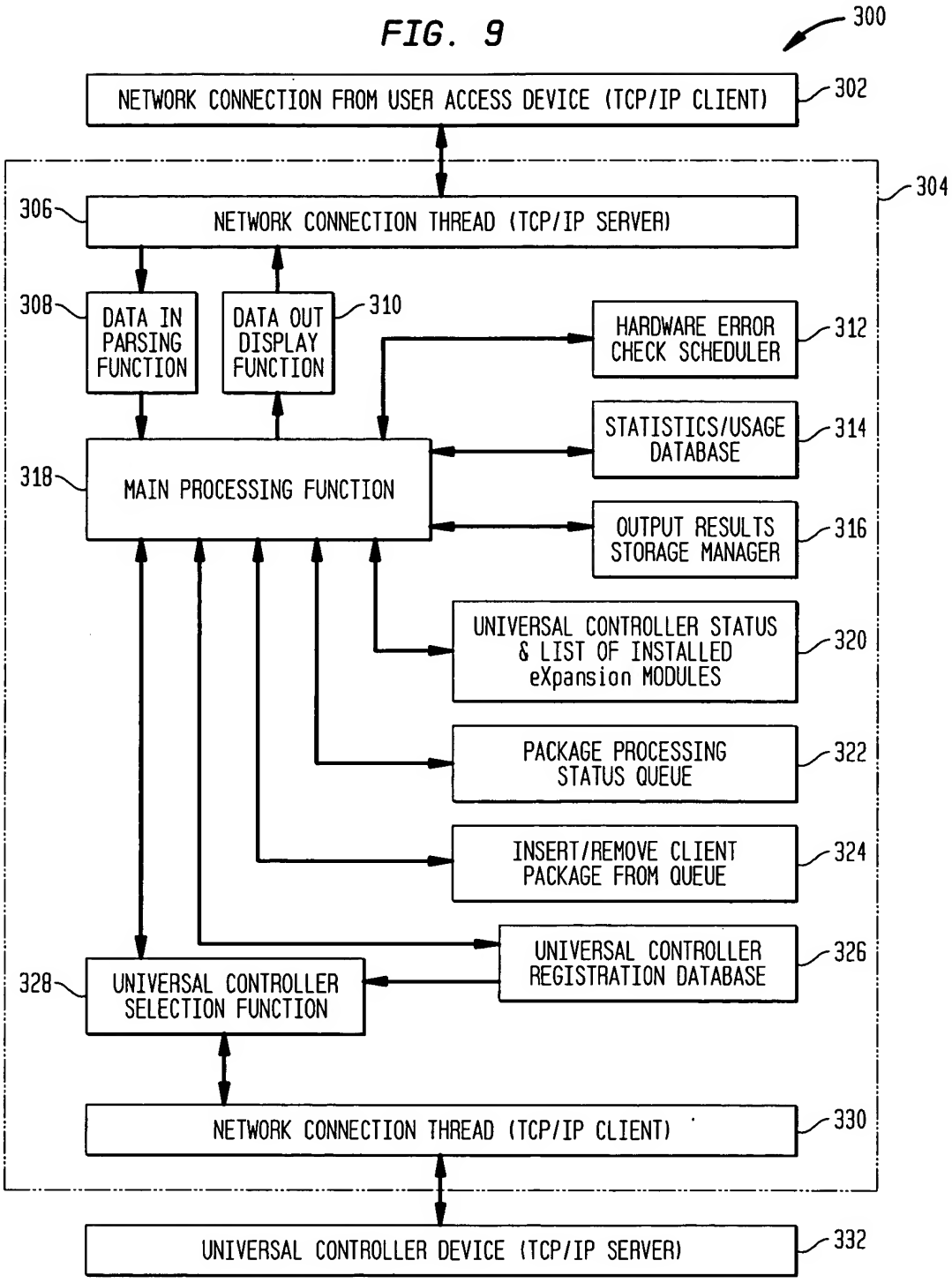

300
NETWORK CONNECTION FROM USER ACCESS DEVICE (TCP/IP CLIENT)
302
304
306
NETWORK CONNECTION THREAD (TCP/IP SERVER)
308
DATA IN PARSING FUNCTION
DATA OUT DISPLAY FUNCTION
310
HARDWARE ERROR CHECK SCHEDULER
312
STATISTICS/USAGE DATABASE
314
318
MAIN PROCESSING FUNCTION
OUTPUT RESULTS STORAGE MANAGER
316
UNIVERSAL CONTROLLER STATUS & LIST OF INSTALLED eXpansion MODULES
320
PACKAGE PROCESSING STATUS QUEUE
322
INSERT/REMOVE CLIENT PACKAGE FROM QUEUE
324
UNIVERSAL CONTROLLER REGISTRATION DATABASE
326
328
UNIVERSAL CONTROLLER SELECTION FUNCTION
NETWORK CONNECTION THREAD (TCP/IP CLIENT)
330
UNIVERSAL CONTROLLER DEVICE (TCP/IP SERVER)
332

FIG. 10

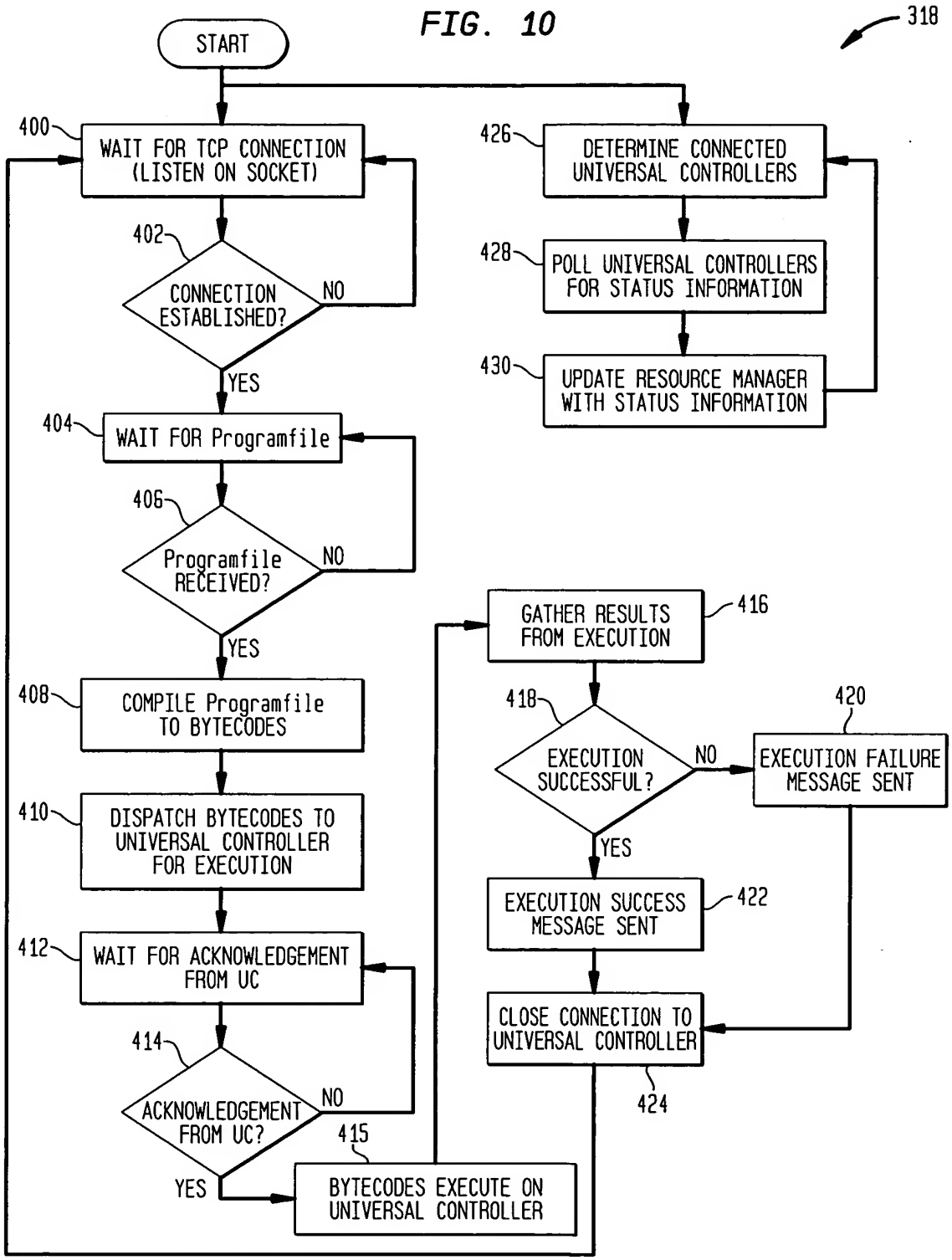

318
START
400
WAIT FOR TCP CONNECTION (LISTEN ON SOCKET)
402
CONNECTION ESTABLISHED?
NO
YES
404
WAIT FOR Programfile
406
Programfile RECEIVED?
NO
YES
408
COMPILE Programfile TO BYTECODES
410
DISPATCH BYTECODES TO UNIVERSAL CONTROLLER FOR EXECUTION
412
WAIT FOR ACKNOWLEDGEMENT FROM UC
414
ACKNOWLEDGEMENT FROM UC?
NO
YES
415
BYTECODES EXECUTE ON UNIVERSAL CONTROLLER
426
DETERMINE CONNECTED UNIVERSAL CONTROLLERS
428
POLL UNIVERSAL CONTROLLERS FOR STATUS INFORMATION
430
UPDATE RESOURCE MANAGER WITH STATUS INFORMATION
416
GATHER RESULTS FROM EXECUTION
418
EXECUTION SUCCESSFUL?
NO
420
EXECUTION FAILURE MESSAGE SENT
YES
422
EXECUTION SUCCESS MESSAGE SENT
424
CLOSE CONNECTION TO UNIVERSAL CONTROLLER

FIG. 11A

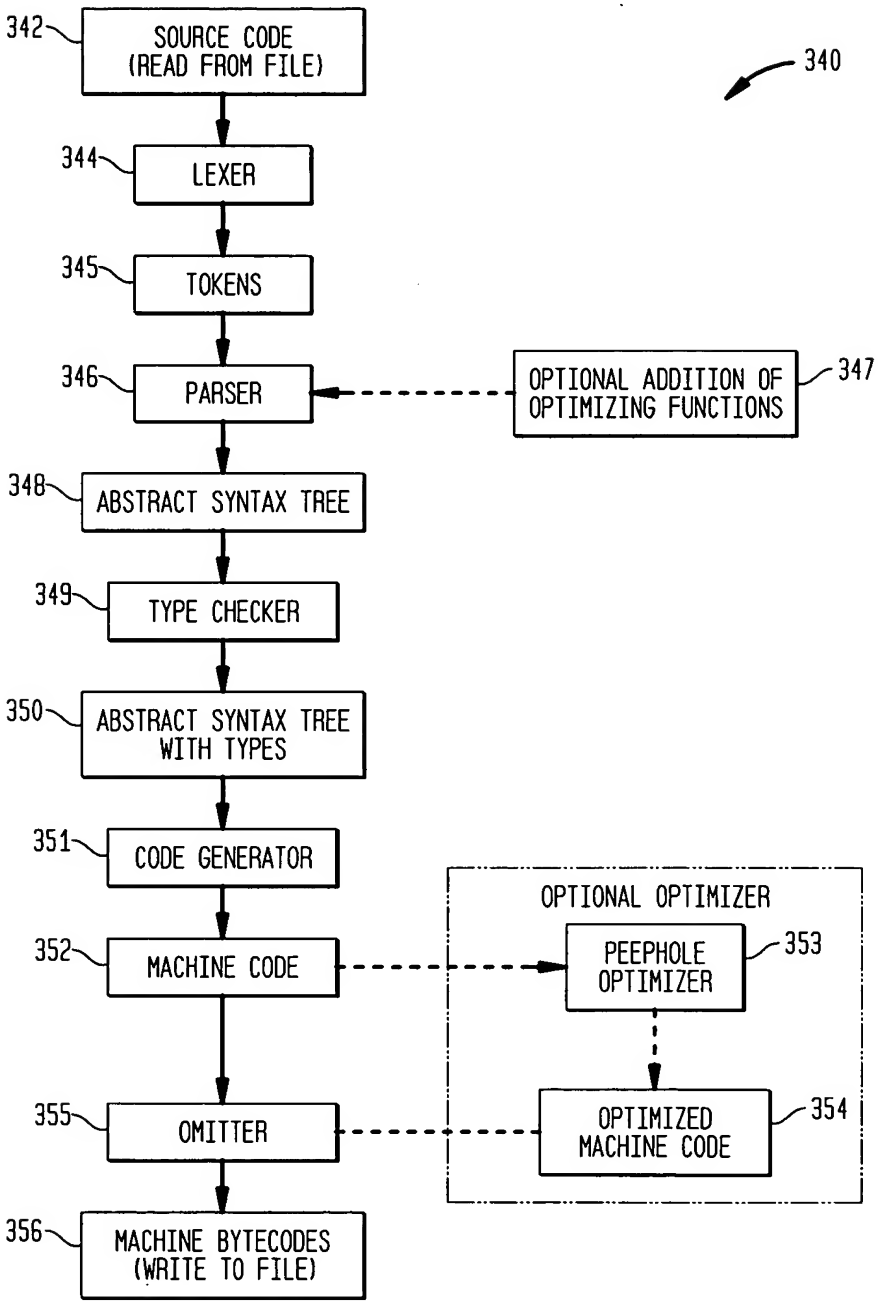

342
SOURCE CODE (READ FROM FILE)
340
344
LEXER
345
TOKENS
346
PARSER
OPTIONAL ADDITION OF OPTIMIZING FUNCTIONS
347
348
ABSTRACT SYNTAX TREE
349
TYPE CHECKER
350
ABSTRACT SYNTAX TREE WITH TYPES
351
CODE GENERATOR
OPTIONAL OPTIMIZER
352
MACHINE CODE
PEEPHOLE OPTIMIZER
353
355
OMITTER
OPTIMIZED MACHINE CODE
354
356
MACHINE BYTECODES (WRITE TO FILE)

FIG. 11B

ucc (HIGH LEVEL)

```
#define    ADC_ADDR    23
#define    CH1         1

void main (void)
{
int x = 78;
int y = 10;

while (y <= 0)  {
      y =
read_adc(ADC_ADDR, CH1);
      }
y = y + x;
}
```

ucASM (LOW-LEVEL)

```
      bipush   78
      istore_1      ;x = 78
      bipush   10
      istore_2      ;y = 10
loop: xmf  23   1   ;read_adc()
      istore_2      ;y=read_adc()
      ifle  loop    ;if y<=0 loop
      iload_2
      iload_1
      iadd          ;y=y+x
      istore_2
      end           ;end of prog
```

BYTECODES

```
1078
3D
1010
3E
F102032301
3E
9E05
1C
1B
60
3E
FF
```

FIG. 12

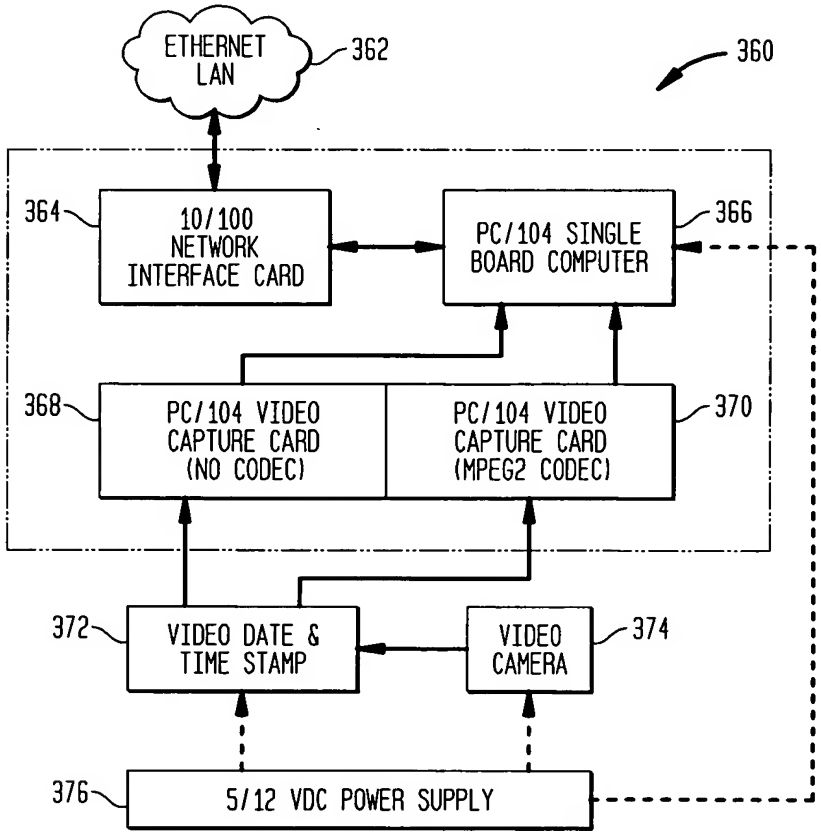


FIG. 13

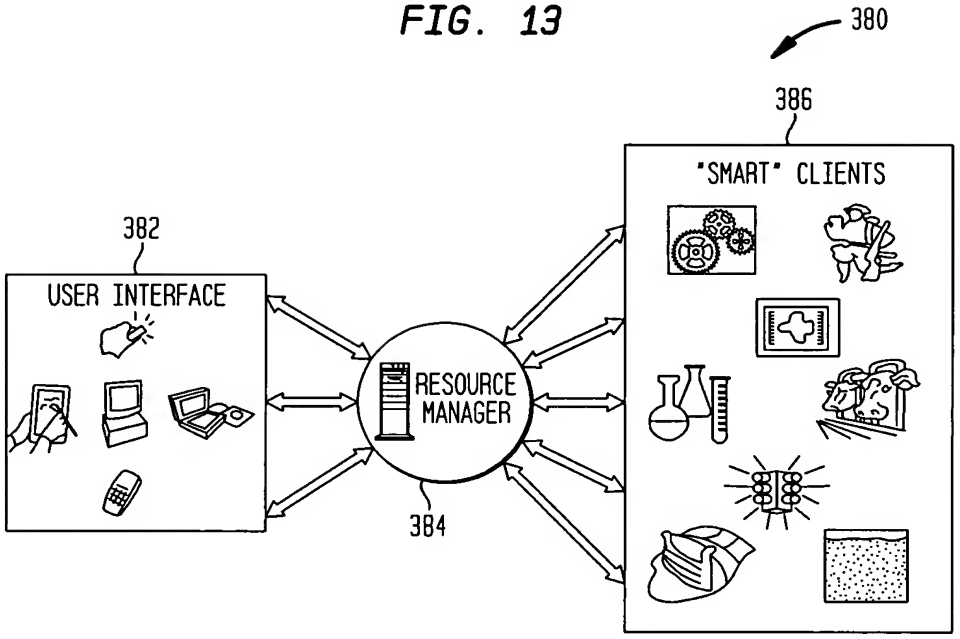

380
386
"SMART" CLIENTS
382
USER INTERFACE
RESOURCE MANAGER
384

FIG. 14

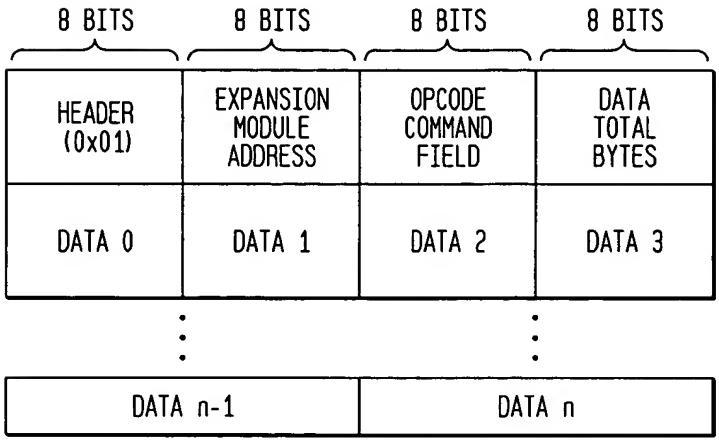

8 BITS
8 BITS
8 BITS
8 BITS
HEADER (0x01)
EXPANSION MODULE ADDRESS
OPCODE COMMAND FIELD
DATA TOTAL BYTES
DATA 0
DATA 1
DATA 2
DATA 3
DATA n-1
DATA n

FIG. 15

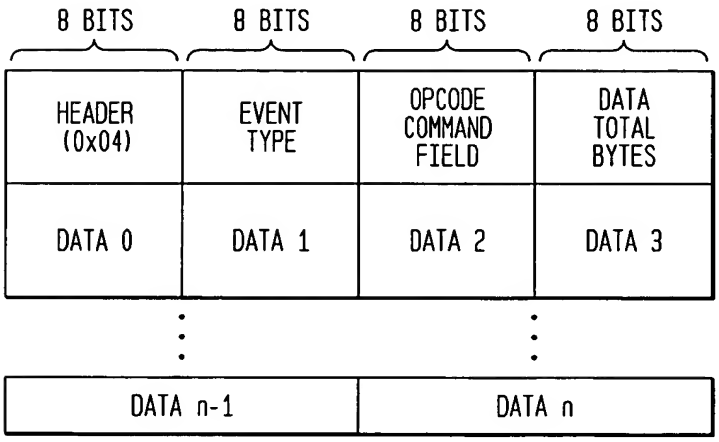

8 BITS
8 BITS
8 BITS
8 BITS
HEADER (0x04)
EVENT TYPE
OPCODE COMMAND FIELD
DATA TOTAL BYTES
DATA 0
DATA 1
DATA 2
DATA 3
DATA n-1
DATA n

FIG. 16

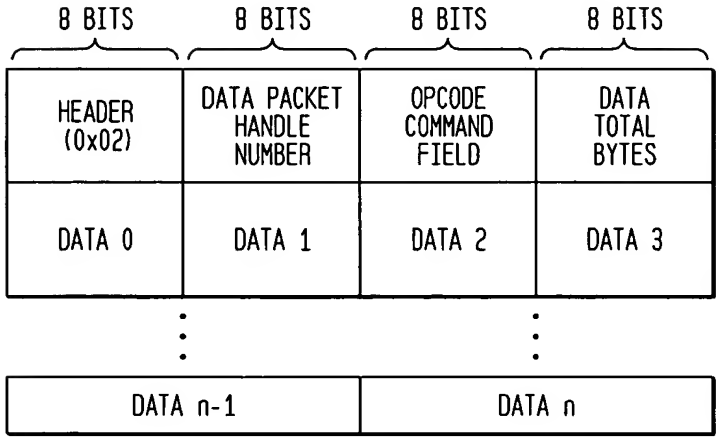

8 BITS
8 BITS
8 BITS
8 BITS
HEADER (0x02)
DATA PACKET HANDLE NUMBER
OPCODE COMMAND FIELD
DATA TOTAL BYTES
DATA 0
DATA 1
DATA 2
DATA 3
DATA n-1
DATA n

FIG. 17

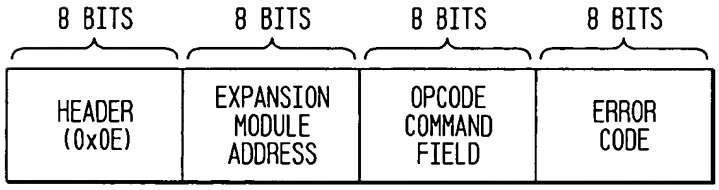

8 BITS
8 BITS
8 BITS
8 BITS
HEADER (0x0E)
EXPANSION MODULE ADDRESS
OPCODE COMMAND FIELD
ERROR CODE

FIG. 18

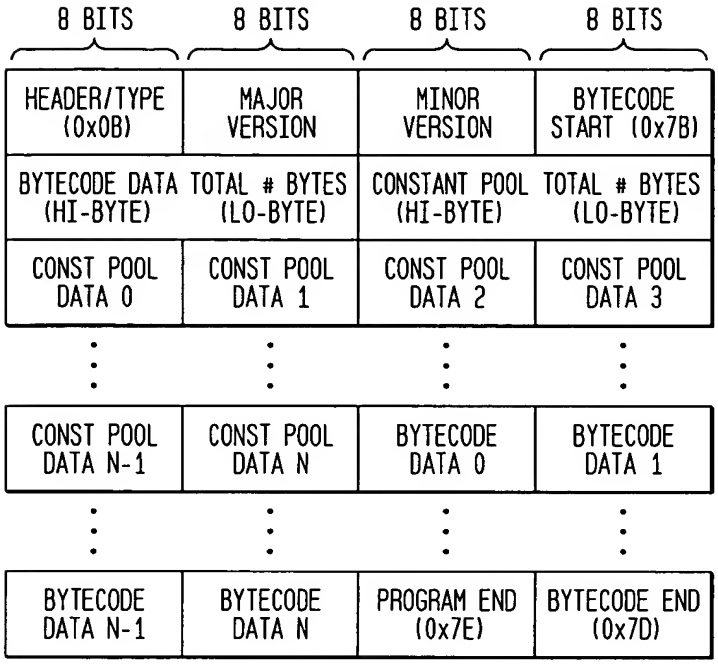
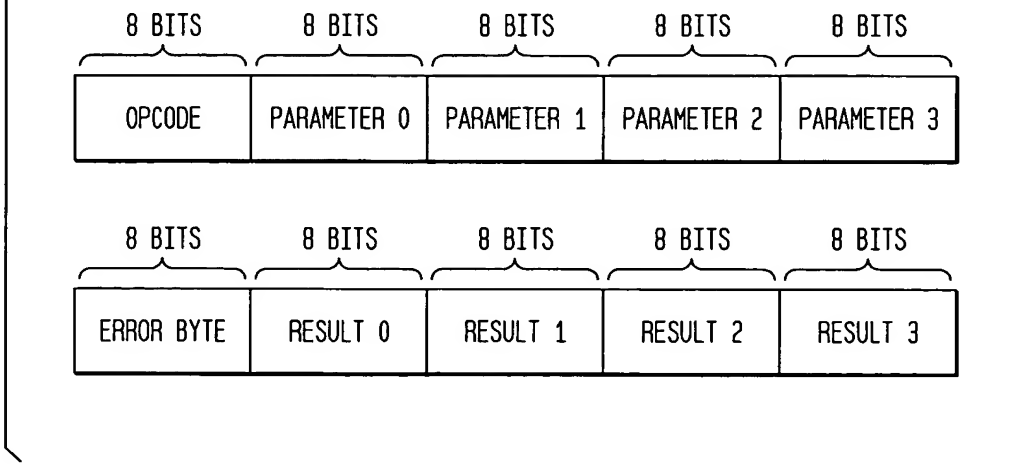

8 BITS
8 BITS
8 BITS
8 BITS
HEADER/TYPE (0x0B)
MAJOR VERSION
MINOR VERSION
BYTECODE START (0x7B)
BYTECODE DATA TOTAL # BYTES (HI-BYTE)
BYTECODE DATA TOTAL # BYTES (LO-BYTE)
CONSTANT POOL TOTAL # BYTES (HI-BYTE)
CONSTANT POOL TOTAL # BYTES (LO-BYTE)
CONST POOL DATA 0
CONST POOL DATA 1
CONST POOL DATA 2
CONST POOL DATA 3
CONST POOL DATA N-1
CONST POOL DATA N
BYTECODE DATA 0
BYTECODE DATA 1
BYTECODE DATA N-1
BYTECODE DATA N
PROGRAM END (0x7E)
BYTECODE END (0x7D)

FIG. 19


8 BITS
8 BITS
8 BITS
8 BITS
8 BITS
OPCODE
PARAMETER 0
PARAMETER 1
PARAMETER 2
PARAMETER 3
8 BITS
8 BITS
8 BITS
8 BITS
8 BITS
ERROR BYTE
RESULT 0
RESULT 1
RESULT 2
RESULT 3